Computer-Aided VLSI System Design
Lab2: Waveform Debugging

TA: 朱怡蓁 r10943012@ntu.edu.tw

Introduction

In this lab, you will learn:

1. How to dump FSDB waveform file through VCS simulator
2. Waveform debugging by using nWave
3. SDF annotation for gate-level simulation after synthesis

Data Preparation

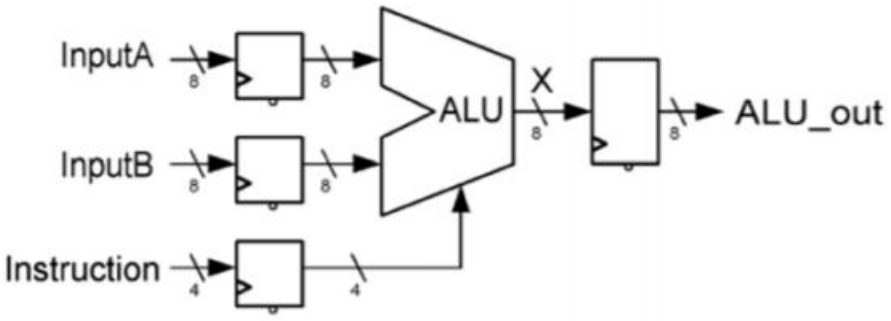
1. Upload your files (Lab2.tar) to your work directory.
2. Decompress Lab2.tar with following command:

```
tar -xvf Lab2.tar
```

3. Lab2 files are shown as below:

Files/Folder	Description
`Lab2_alu.v`	RTL code of ALU
`Lab2_test_alu.v`	Testbench for ALU
`Lab2_alu_run.f`	File list for RTL simulation
`Lab2_alu_s.v`	Gate-level netlist of ALU
`Lab2_alu_run_s.f`	File list for gate-level simulation
`Lab2_alu_s.sdf`	SDF file for gate-level timing annotation

4. The circuit diagram of the ALU used for this lab is illustrated as follows:



5. Enter Lab2 directory:

```
cd lab2
```

Environment Setup

1. Source the default **cshrc** file:

```
source /usr/cad/synopsys/CIC/vcs.cshrc
source /usr/spring_soft/CIC/verdi.cshrc
```

Lab 2.1. Generating the VCD & FSDB Waveform

Value Change Dump (VCD) format:

- Indigenously supported by most simulators
- Using ASCII text for waveform recording (larger file size)

```
$dumpfile("filename.vcd");
$dumpvars;
```

Fast Signal Database (FSDB) format:

- Defined by Verdi debugging system
- More compact format, small file size

```
$fsdbDumpfile("filename.fsdb");
$fsdbDumpvars;
```

1. Run the RTL simulation using VCS:

```
vcs -full64 Lab2_test_alu.v Lab2_alu.v +v2k -R
```

or

```
vcs -full64 -f Lab2_alu_run.f +v2k -R
```

2. You would only see the text telling you "congratulations" but not knowing what's going on inside the circuit. Now we would like to generate VCD file.

   Please open the testbench `Lab2_test_alu.v`, add following lines in the `initial` block in testbench to generate vcd file:

```
$dumpfile("Lab2_alu.vcd");
$dumpvars;
```

3. Now re-run the simulation, note we should add "`-debug_access+all`" to enable vcd file dumping:

```
vcs -full64 -f Lab2_alu_run.f +v2k -R -debug_access+all
```

4. Please check if there exists a file called "`Lab2_alu.vcd`"
5. Now we would like to generate FSDB file. Please open the testbench again.

6. Add following lines in the `initial` block in `Lab2_test_alu.v`, you can also remove the two lines added in step 2:

```
$fsdbDumpfile("Lab2_alu.fsdb");
$fsdbDumpvars;
```

7. Now re-run the simulation, note we should add "`-debug_access+all`" to enable FSDB file dumping:

```
vcs -full64 -f Lab2_alu_run.f +v2k -R -debug_access+all
```

8. Please check if there exists a file called "`Lab2_alu.fsdb`"

Lab 2.2. Using nWave for Waveform Debugging

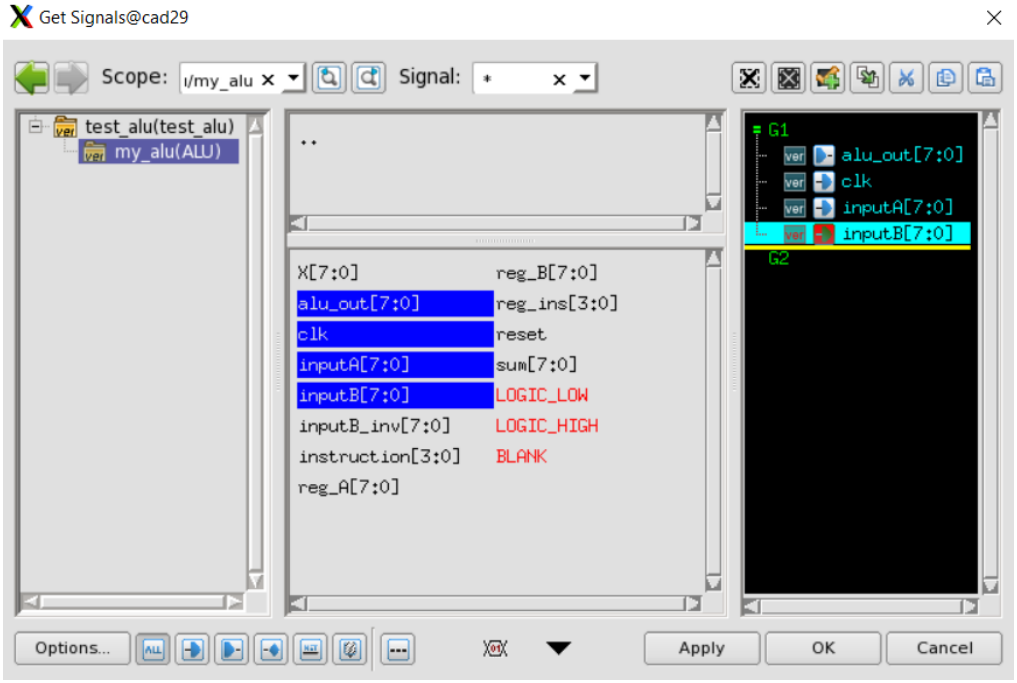
1. Start nWave. "&" enables you to use the terminal while nWave is running in the background:

```
nWave &
```

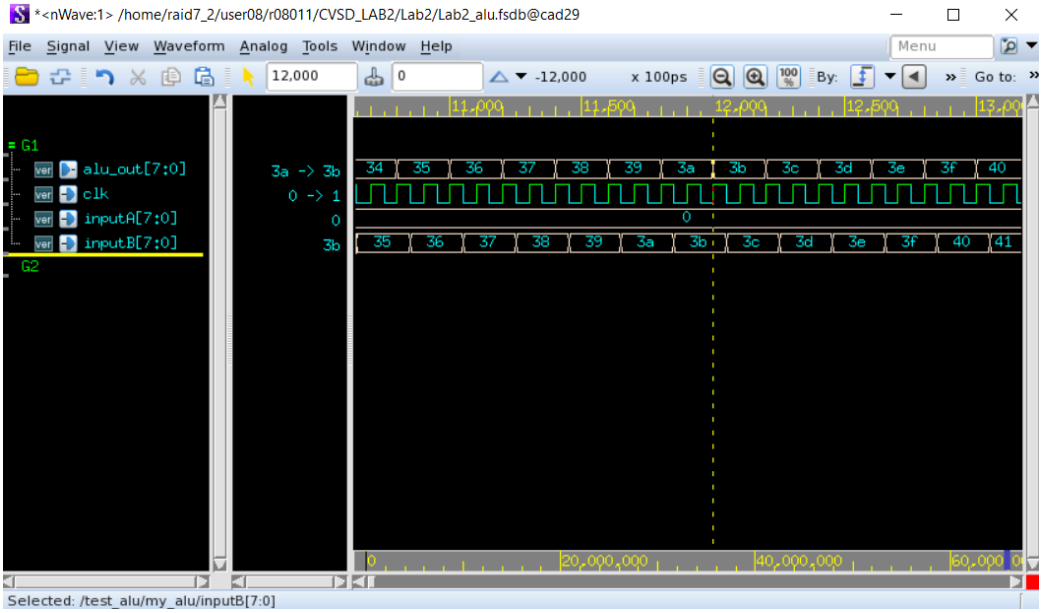
2. Open the VCD file we obtained. VCD file will be translated to FSDB file, you can either open the VCD file using GUI or run:

```
nWave Lab2_alu.vcd &
```

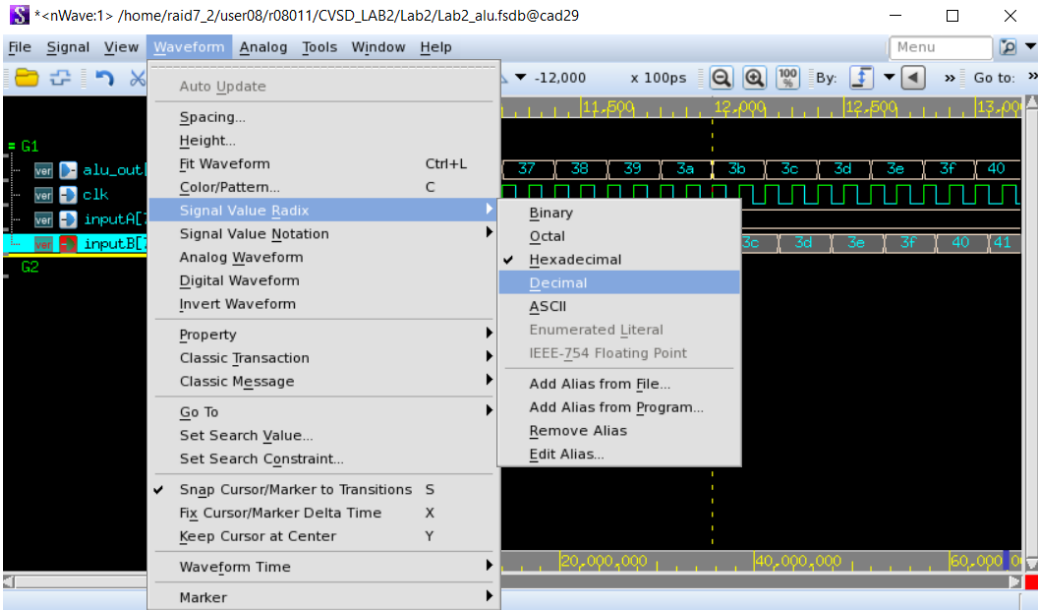
3. Choose the signals we are interested in.



4. View the waveform:



5. Change the radix and sign representation:
   - "Waveform" – "Signal Value Radix" – "Decimal"
   - "Waveform" – "Signal Value Notation" – "Signed 2's Complement"



Lab 2.3. Gate-level Simulation

1. After synthesizing `Lab2_alu.v`, we can derive the gate-level netlist file "`Lab2_alu_s.v`", which is already provided in this lab.

   Try this command for gate-level simulation:

   ```
   vcs -full64 +v2k -R +neg_tchk -debug_access+all Lab2_test_alu.v Lab2_alu_s.v
   /home/raid7_2/course/cvsd/CBDK_IC_Contest_v2.5/Verilog/tsmc13_neg.v
   ```

   or

   ```
   vcs -full64 -f Lab2_alu_run_s.f +v2k +neg_tchk -R -debug_access+all
   ```

2. You will see a lot of warnings about "timing violation." That's because the simulator does not know about the propagation delay of each gate. So the simulator cannot combine the delay information with the "tsmc13_neg.v" to check if your design can run at the clock frequency defined in the testbench.

   Please open the testbench `Lab2_test_alu.v`, and uncomment this line:

   ```
   $sdf_annotate("Lab2_alu_s.sdf", my_alu);
   ```

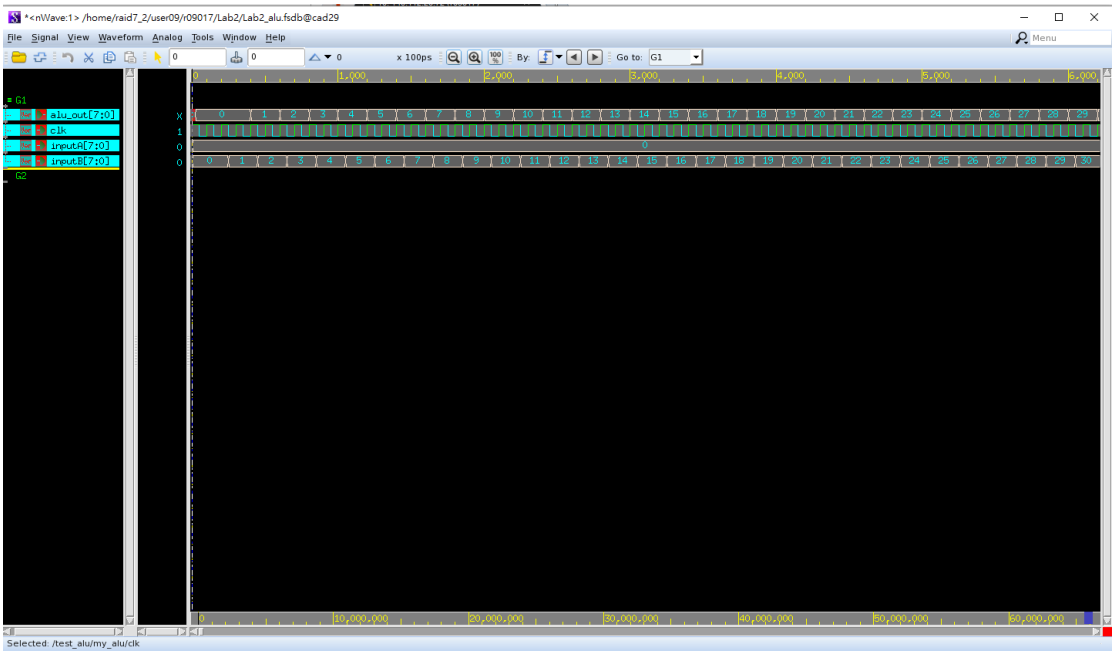
   This line annotates the timing information (.sdf) of the netlist.

3. Save the modified testbench and re-run step 1.
4. During the simulation, there is no timing violation anymore.
5. Open the waveform file to see how different it is between RTL waveform and gate-level waveform.

Checkpoints

Please check with TAs before leaving this lab to make sure the following goals are accomplished and to get credits.

1. Please generate the FSDB waveform.
2. Open nWave and take the snapshots of alu_out, clk, inputA and inputB.



3. Fix the timing violations in gate-level simulation.
4. Take a snapshot after passing gate-level simulation.

```
   ***   $sdf_annotate() version 1.2R
   ***    SDF file: "Lab2_alu_s.sdf"
   ***    Annotation scope: test_alu.my_alu
   ***    No MTM selection argument specified
   ***    No SCALE FACTORS argument specified
   ***    No SCALE TYPE argument specified
   ***    MTM selection defaulted to "TOOL_CONTROL":
              (+typdelays compiled, TYPICAL delays selected)
   ***    SCALE FACTORS defaulted to "1.0:1.0:1.0":
   ***    SCALE TYPE defaulted to: "FROM_MTM"
   ***    Turnoff delay: "FROM_FILE"
   ***    Approximation (mipd) policy: "MAXIMUM"

   ***    SDF annotation begin: Fri Sep 22 14:39:57 2023


SDF Info: +pulse_r/100, +pulse_e/100 in effect

          Total errors: 0
          Total warnings: 0
   ***    SDF annotation completed: Fri Sep 22 14:39:57 2023
```

```
rm -f _cuarc*.so _csrc*.so pre_vcsobj_*.so share_vcsobj_*.so
if [ -x ../simv ]; then chmod a-x ../simv; fi
g++  -o ../simv      -rdynamic  -Wl,-rpath='$ORIGIN'/simv.daidir -Wl,-rpath=./simv.daidir -Wl,-rpath=/usr/cad/synopsys/vcs/2022.06/li
nux64/lib -L/usr/cad/synopsys/vcs/2022.06/linux64/lib  -Wl,-rpath-link=./  /usr/lib64/libnuma.so.1   objs/amcQw_d.o   _119606_archive
_1.so objs/udps/D2wHf.o objs/udps/uYEPC.o objs/udps/vCfas.o objs/udps/IEZrF.o objs/udps/U7Vwg.o objs/udps/i2VqJ.o objs/udps/CjLsY.o o
bjs/udps/sk4QJ.o objs/udps/exIG1.o objs/udps/gSqMj.o objs/udps/Cpxa2.o objs/udps/AubIW.o  SIM_l.o      rmapats_mop.o rmapats.o rmar.o
 rmar_nd.o  rmar_llvm_0_1.o rmar_llvm_0_0.o          -lvirsim -lerrorinf -lsnpsmalloc -lvfs    -lvcsnew -lsimprofile -luclinative /u
sr/cad/synopsys/vcs/2022.06/linux64/lib/vcs_tls.o   -Wl,-whole-archive  -lvcsucli    -Wl,-no-whole-archive       _vcs_pli_stub_.o
/usr/cad/synopsys/vcs/2022.06/linux64/lib/vcs_save_restore_new.o /usr/cad/synopsys/verdi/cur/share/PLI/VCS/LINUX64/pli.a -ldl  -lc -l
m -lpthread -ldl
../simv up to date
Chronologic VCS simulator copyright 1991-2022
Contains Synopsys proprietary information.
Compiler version T-2022.06_Full64; Runtime version T-2022.06_Full64;  Sep 22 14:33 2023
Doing SDF annotation ...... Done

Congratulations!! Your Verilog Code is correct!!

$finish called from file "Lab2_test_alu.v", line 129.
$finish at simulation time          6558735000
           V C S   S i m u l a t i o n   R e p o r t
Time: 6558735000 ps
CPU Time:      1.650 seconds;       Data structure size:   0.2Mb
Fri Sep 22 14:33:51 2023
CPU time: 4.155 seconds to compile + .499 seconds to elab + 1.810 seconds to link + 1.728 seconds in simulation
[r10012@cad27 lab2]$
```

Submission

1. **Due Tuesday, Oct. 3, 19:00**

2. Take the snapshot of the result shown in previous section and submit to NTU COOL in pdf format.